2

TECHNICAL SUPPORT TASK REPORT FOR THE MODERNIZATION OF DEFENSE LOGISTICS STANDARD SYSTEMS

Volume I: Prototype Test Report

Report DL702R1

May 1991

William T. James, III

With
Christo G. Andonyadis
John S. Doby
John Lycas
Don Wilson

Prepared pursuant to Department of Defense Contract MDA903-90-C-0006. The views expressed here are those of the Logistics Management Institute at the time of issue but not necessarily those of the Department of Defense. Permission to quote or reproduce any part – except for Government purposes – must be obtained from the Logistics Management Institute.

LOGISTICS MANAGEMENT INSTITUTE
6400 Goldsboro Road
Bethesda, Maryland 20817-5886

REPORT DOCUMENTATION PAGE	*Form Approved* *OPM No. 0704-0188*

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE May 1991	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE Technical Support Task Report for the Moderization of Defense Logistics Standard Systems – Volume I: Prototype Test Report	5. FUNDING NUMBERS C MDA903-90-C-0006 PE 0902198D
6. AUTHOR(S) William T. James, III with Christo G. Andonyadis, John S. Doby, John Lycas, Don Wilson	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Logistics Management Institute 6400 Goldsboro Road Bethesda, MD 20817-5886	8. PERFORMING ORGANIZATION REPORT NUMBER LMI-DL702R1
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Logistics Standard Systems Division 6301 Little River Turnpike, Suite 210 Alexandria, VA 22312	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION/AVAILABILITY STATEMENT A: Approved for public release; distribution unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This volume describes the MODELS Technical Support task activities. It identifies the problems to be solved, the planned approach to their solution, the conclusions reached, and the recommendations for a final implementation. From this study, operational lessons were learned and specifications were drawn; these are the subjects of Volumes II and III of this report.

14. SUBJECT TERMS MODELS, EDI Translation, LGN, CLGN, Prototype	15. NUMBER OF PAGES 70
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	18. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

PREFACE

This technical report, in three volumes, is the final report covering more than 2 years of technical activity supporting the Modernization of Defense Logistics Standard Systems (MODELS) project. The supporting activities included developing translation tables and the table-driven software for converting current fixed-length logistics data formats into new variable-length transaction equivalents. They also included designing and testing prototype hardware and software platforms that support transaction interchange between logistics sites.

This volume, *Prototype Test Report*, is Volume I of the series. It is an overview describing the task's purpose, results, conclusions, and recommendations from the viewpoint of four major support activities:

- Prototype logistics gateway node (LGN) construction and testing
- Interconnection and control of telecommunicating LGNs
- Electronic data interchange transaction translation and testing
- Network performance simulation and cost modeling.

Volume II, *Logistics Gateway Node Prototype Construction and Operation*, details the construction and operation of the prototype LGNs developed for the support task. The document is written from the perspective of software performance. It is provided to assist in understanding and implementing the technical specification developed for the LGN.

Volume III, *Logistics Gateway Node Technical Specification* presents the performance requirements for an LGN and central LGN (CLGN) interconnected within a homogeneous network of LGNs under CLGN control. The specification's purpose is to describe the technical capabilities necessary for a network of deployed LGNs to meet the functional capability called for in OSD directives.

LMI

Executive Summary

TECHNICAL SUPPORT TASK REPORT FOR THE MODERNIZATION OF DEFENSE LOGISTICS STANDARD SYSTEMS

The Modernization of Defense Logistics Standard Systems (MODELS) will substantially increase the responsiveness and flexibility of DoD logistics by redesigning and upgrading the Defense Logistics Standard Systems (DLSS). The substitution of revised variable-length electronic data interchange (EDI) transactions for today's 80-column formats represents a significant advance in logistics information management practices.

To take advantage of the variable-length transactions, the hardware and software used in the various information processes need to be modernized. The major technical development supporting MODELS transactions is a small logistics gateway node (LGN) processor.

An LGN performs technical modifications to transactions entering and leaving its logistics site. LGNs will be necessary to permit simultaneous operation of current DLSS-capable and EDI-capable logistics activities during the protracted transition to a system composed solely of EDI-capable sites. LGN operations include the following:

- Translating between DLSS and EDI formats
- Compressing and decompressing transmitted data
- Safeguarding transactions by encrypting and decrypting
- Editing and routing, to a limited extent.

To ensure effective technical support in exchanging EDI-formatted logistics transactions, we recommend to OSD the following:

- DLSS logistics traffic should be converted from 80-column messages on the antiquated Automatic Digital Network service to compressed EDI formats using the Defense Data Network or its directed equivalent. Data compression between homogeneous sending and receiving LGNs can offset the increase in transaction size resulting from EDI formatting.

- A standard DoD table-driven translation software should be developed to convert between existing DLSS and EDI formats. Further, the driving tables should employ the specific translation logic developed and thoroughly tested during the MODELS prototype testing project.

- The Defense Automatic Addressing System Office should upgrade its central configuration to support a distributed network consisting of interconnected LGNs. Deployed LGNs can provide effective and economical support for exchanging EDI and DLSS transactions during the transition period.

- The Defense Automatic Addressing System Office should develop an internal processing capability for EDI-formatted transactions. Retranslation into the 80-column format at the central telecommunications hub represents an insupportable increase in workload at central sites, as well as a loss of logistics data.

A network of LGNs constructed in accordance with these recommendations offers the following improvements over the current system:

- DoD trading partners will be able to phase in their implementation of EDI transactions without waiting for across-the-board readiness among all Service and Agency activities.

- Being able to exchange logistics transactions through LGN contingency operation will eliminate the current two-site telecommunications hub's vulnerabilities.

- The functional capability called for in OSD directives will be able to be met via a low-cost investment of approximately $10,000 for each site. The number of sites will depend on the implementation rate of the Corporate Information Management initiative and may be as many as 100 sites.

CONTENTS

	Page
Preface	iii
Executive Summary	v
Chapter 1. Introduction	1- 1
Technical Issues	1- 3
LGN Prototype Objective	1- 4
Software Objective	1- 4
Telecommunications Objective	1- 4
Processing Distribution Objective	1- 5
Functional Issues	1- 5
Structure of Volume	1- 6
Chapter 2. The Approach	2- 1
Prototype LGN Overview	2- 1
Phase I	2- 1
Phase II	2- 2
Phase III	2- 2
Translation Tables Overview	2- 3
Tables DLSS-to-EDI	2- 4
Tables EDI-to-DLSS	2- 4
Test Transaction Formats	2- 5
Translation Software Overview	2- 5
Field Test Overview	2- 5
Simulation and Cost Models	2- 7
Chapter 3. Prototype Test	3- 1
The Field Test Experience	3- 1
Download Process	3- 1
Script Program Processing	3- 2
Simulation and Cost Models	3- 3
Input Values and Processing Compromises	3- 4
Volume Threshold	3- 4
Time Scales	3- 5

CONTENTS (Continued)

	Page
Treatment of DAAS	3- 5
LGN Site Design	3- 6
Chapter 4. Conclusions	4- 1
Prototype Construction and Operation	4- 1
LGN Functionality	4- 1
Multitasking	4- 2
Data Compression	4- 2
Transaction Editing	4- 3
Transaction Routing	4- 3
Reduction of DAAS Vulnerability	4- 5
Translation Table Construction	4- 5
Translation Software	4- 7
Prototype Field Test	4- 8
Host Interconnection	4- 8
Multitasking Capability	4- 8
Telecommunication Alternatives	4- 9
Validation of Transaction Design	4- 9
Simulation and Cost Model	4-10
Simulation Model Parameters	4-10
Cost Model Parameters	4-11
Cost Model Results	4-11
Chapter 5. Recommendations	5- 1
CLGN Provisions	5- 1
Deployed LGN Provisions	5- 3
Transaction Editing Provisions	5- 4
Data Transmission Provisions	5- 5
Translation Table Construction	5- 6
Prototype Translation Software	5- 6
Appendix A. The Prototype Test Plan	A-1 – A- 8
Appendix B. Field Test Examples	B-1 – B- 4
Appendix C. Cost Model Results	C-1 – C- 5
Appendix D. Acronyms	D-1 – D- 4

CHAPTER 1
INTRODUCTION

The Defense Logistics Standard Systems (DLSS) are a set of rules, procedures, and data standards first created in the early 1960s to support DoD's consolidation of functions. Before the Defense Supply Agency, now called the Defense Logistics Agency (DLA), was established in 1961, each Military Service took quite literally its mission to raise, equip, and feed an Army, Navy, or Air Force. As a result, each Service competed against the others in procuring items; was unable to order economic quantities; duplicated expensive storage, transportation, and maintenance activities; and was unaware that the other Services might have excess stocks of an item desperately needed.

To increase efficiency and effectiveness, the Secretary of Defense established the policy of having a single item manager and founded DLA. This meant that logistics personnel in one DoD organization might order, receive, and ship items to four different organizations (Army, Navy, Air Force, and DLA). Since each organization had different forms and procedures, logisticians were inundated with paper and with unfamiliar rules for filling out the forms.

To prevent total chaos, the Services and DLA agreed upon the DLSS. Most of the standards were automated using 1960s automated data processing (ADP) technologies. Although there have been improvements and enhancements, the DLSS are still based on the precepts set 30 years ago.

In the intervening years, further consolidations and additional coordination between the Services and DLA have occurred. More consolidations and even closer coordination are expected in the near term. The DLSS are unquestionably not adequate to support today's logistics structure, and they are becoming more antiquated daily. During the 1980s, the Office of the Secretary of Defense (OSD) came to realize that, like a 2-lane bridge on a busy highway, the DLSS had to be modernized. Furthermore, the Services had built nonstandard "pontoon bridge" procedures that, once again, inundated logisticians with unfamiliar forms and rules for numerous procedures. The answer to these problems was the Modernization of Defense Logistics Standard Systems (MODELS) project.

The MODELS program is managed by the Director of the Defense Logistics Standard Systems Division (DLSSD). The Logistics Management Institute (LMI) was chosen to do the functional analysis and transaction development and to provide initial technical support for developing the MODELS program. The Defense Automatic Addressing System Office (DAASO) will be responsible for the ongoing technical support of the modernized system.

The MODELS program is designed to (1) increase the overall effectiveness of logistics management functions, (2) improve the flexibility and capabilities of DoD inter-Service/Agency operation, and (3) improve management information communications by increasing the use of standard data element definitions, common transaction formats, and electronic communication protocols. These goals are to be attained in such a way that the Services and Agencies will not be inhibited in improving their own logistics activities. The program has two major thrusts:

- A functional reexamination of the data that logisticians need in order to procure, supply, maintain, transport, and dispose of materiel
- A technical upgrading of the data standards and software, hardware, and communication technologies required to exchange the data effectively and efficiently.

Overall, MODELS is to provide the standard policies, procedures, and transaction formats to facilitate effective communication of vital logistics information in and among Service and Agency ADP systems. This volume of a three-volume report focuses on the issues and problems affecting the technical upgrade thrust of MODELS and on their recommended solutions.

The Technical Working Group (TWG), with representatives from the Office of the Assistant Secretary of Defense (Production and Logistics), DLA, the Services, LMI, DAASO, and DLSSD, had the following objectives in mind for the system that would result from the current effort:

- It must provide a transition from present operations to the new technical solution (this can be likened to rebuilding a bridge without ever closing it to traffic).
- It must be cheaper and easier to operate and maintain.

- It must be compatible with DoD technical architecture (i.e., it must use open systems, standard communications, and available hardware and, to the extent feasible, commercial software).

The TWG members met several times to discuss the technical solution. They determined that, although the ingredients of that solution were well-understood and operating in different organizations, the best way to validate it would be to build and test a prototype. The TWG's prototype goals and objectives are documented in LMI's prototype test plan, which is summarized in Appendix A.

The primary emphasis of the technical tasking was to define and develop the processing capabilities necessary to fulfill fundamental technical support functions. These capabilities are to be provided by a new architecture involving the use of computer hardware and software in the form of logistics gateway nodes (LGNs).

TECHNICAL ISSUES

The LGN concept ties together several proven technologies:

- Automated translation of fixed-length transactions into electronic data interchange (EDI) format, using table-driven translation software
- Transmission of variable-length transactions using a packet-switched telecommunications wide area network (WAN)
- Use of American National Standards Institute (ANSI) standard EDI transaction sets to exchange business information.

With the basic elements of an exchange mechanism already well proven, it is the size and scope of DoD's logistics interests that provide the challenge. To date, no system has been compiled whose constituents meet current DoD requirements.

Today, EDI transactions normally are exchanged between two (or, sometimes, a few more) industrial trading partners. But trading partners among DoD logistics activities number in the hundreds. Commercial EDI translators are readily available for individual use or through a third-party service agreement. However, a prerequisite for using them is that data from a host system must be arranged in an unambiguous "flat-file" format. Throughout nearly 30 years of use and extension, the standard 80-character transaction set has evolved into over 425 separate definitions applying to just 80 data positions. A major portion of the DLSS-to-EDI translation job is to decipher unambiguously the logical echelons arising from such multiple use of data element positions for placement into an array.

LGN Prototype Objective

Since there has been no existing MODELS LGN, the problem from the outset was to define as exhaustive a prototype model as could be built and tested within limited time and funding. It was decided that the prototype should demonstrate state-of-the-art technology required in an operational model (multitasking, for example). The solution was to permit an LGN baseline requirement to be inferred by testing a surrogate prototype with significantly fewer features than those demanded in the ultimate, production version.

Software Objective

Processing the convolutions in the DLSS logic – in contrast to performing the usual, relatively simple task of retrieving data from fixed positions in an array – could prove prohibitively time consuming. A comprehensive test was needed to determine realistic LGN throughput targets within an affordable small-machine strategy.

To verify the feasibility of translating between DLSS and EDI formats in a small-machine environment, translation tables and software to interpret them had to be developed for the prototype. It was decided that such translation software should employ off-the-shelf routines, preferably capitalizing on the wealth of available support for the C language and the MicroSoft Disc Operating System (MS-DOS) environment. The advantage of software tailored specifically for the prototype is that it can be written quickly, altered as vital lessons are learned, and discarded after the test. Indeed, many-times-patched software would be difficult to maintain in an operational mode and should be discarded upon completion of its task.

Telecommunications Objective

Over the years, the exchange of DLSS documents (transactions) has escalated to the point where roughly 40 percent of the capacity of the aging Automatic Digital Network (AUTODIN) telecommunications system is consumed by logistics traffic. Although its modern counterpart, the Defense Data Network (DDN), theoretically can be sized to accommodate virtually any volume of traffic, it is not clear how cost-effectively or how efficiently DDN will be able to meet increasing workloads from logistics activities making the transition to EDI. The prototype test addressed the telecommunications issue to the extent that a limited LGN field test could allay the telecommunications concerns.

Processing Distribution Objective

From the outset, the question of "how many LGNs at how many operational sites" produced animated debate, with MODELS participants expressing strong and varied opinions. Depending upon the point of view being articulated, one could infer quite dissimilar notions about how MODELS transaction exchanges would be supported. The following are some of the basic issues:

- Appropriate level of distribution of LGN equipment
- Baseline functionality for deployed LGNs
- Role of the Defense Automatic Addressing System (DAAS) in a network of interconnected LGNs
- Cost of transmitting logistics traffic over DDN in EDI format versus cost of transmitting present fixed-length transactions via AUTODIN
- Impact of direct LGN to-LGN transmission using DDN on daily workload distribution
- Efficacy of direct LGN-to-LGN transmission as opposed to using a central processing hub.

The information required for resolving all of these kinds of questions transcends the capacity of a limited field test. Therefore, the technical project developed and employed simulation and cost models to augment the field test results. Specifically, these models provided a better understanding of the degree to which distributed processing would be appropriate and economical.

FUNCTIONAL ISSUES

To meet the basic requirements that have been outlined, the current system of standard data elements was redesigned by a cooperative effort involving LMI, DLSSD, and the Functional Working Group (whose membership parallels that of the TWG). The result is a drastic reduction in the number of transactions, replacing approximately 425 distinct document identification code (DIC) fixed-length transactions with 56 equivalent variable-length transactions. The 56 MODELS transactions are designed to give logisticians and managers more accurate and timely information without loss of functionality. To ensure the availability of standard communication formats, ANSI ASC (Accredited Standards Committee) X12 EDI formats form the basis for MODELS transactions.

A great benefit of the change in format derives from the potential for exchanging additional logistics information within the defined transactions, facilitating future alterations in standard transactions in response to changing business needs.

To benefit from MODELS, the Services and Agencies need a standard means for interchanging logistics traffic. Moreover, communication between EDI-capable and non-EDI activities will be necessary for an extended transition period. The MODELS technical tasking was instituted to develop and test a prototype LGN able to meet these requirements.

STRUCTURE OF VOLUME

The remainder of this volume describes the approach to the task (Chapter 2), outlines the prototype test activities (Chapter 3), details conclusions drawn from the study (Chapter 4), and offers recommendations based on this work (Chapter 5). Appendix A is the summary of the prototype test plan previously mentioned, while Appendix B provides examples of the field tests. Appendix C presents cost model results. Appendix D defines the acronyms used.

CHAPTER 2

THE APPROACH

PROTOTYPE LGN OVERVIEW

The prototype LGN was implemented on a 20 mega-hertz (Mhz) 386 microprocessor platform in an MS-DOS environment. This choice was made largely on the basis of the rich support available for both software and hardware. The LGN was designed in three separate, progressive phases because such an approach would (1) provide early use of a test vehicle without waiting for planned capabilities of later versions and (2) allow modification in the invention and construction of later-phase LGNs to profit from lessons learned from experience with earlier versions. Thus, the better understood elements of the technical support task could proceed unimpeded by the fact that some processes – whose constituents might be only guesses in the beginning – had not yet been developed.

Phase I

The role of the Phase I starting model was the performance of standalone translation and staging of transactions. This LGN had only translation and limited executive control software. It was a test platform designed to exercise and verify early-version Military Standard Requisition and Issue Procedures (MILSTRIP) translation tables.

The translation methodology for the test began in the DLSS-to-EDI direction. Then, adding retranslation tables – from EDI back to DLSS – permitted conversion in that direction to be tested. To ensure a rapid start, the initial version of the translator employed a slow but user-friendly Paradox data base and script language. The phased approach allowed remaining DLSS tables to be introduced as early as Phase I or in later stages, depending on the rate of LGN development.

Phase II

The second-generation prototype LGN incorporated all the basic elements of an operational LGN network. These include the following features:

- An improved and faster C-language translator
- Asynchronous WAN telecommunications between LGNs
- Direct connection or connection via local area network (LAN) with test-site host computer systems
- Compression and decompression routines
- LGN subordination to central LGN (CLGN) control
- Rudimentary table look-up addressing for transaction routing
- Continuously flowing bi-directional transaction processing.

The Phase II LGN had functional capabilities sufficient to explore all major operational issues without further embellishment in its design. The purpose of testing this second LGN version was to generate data for establishing baseline functional requirements for an operational LGN and for its design specification (see Volume III of this report).

Phase III

The third prototype LGN was developed to approximate more closely the realistic operational state anticipated for a large number of deployed LGNs, their controlling CLGN, and the interconnecting network.

Internal processing for the Phase III model was based on an ability to perform multitasking. Multitasking in the LGN permits automatic, continuous workload flow while also allowing prioritization of tasks and current-task interruption as needed. In the Phase III model, multitasking performance was approximated by using DESQview, an off-the-shelf program designed to emulate multitasking in MS-DOS microprocessors. Because MS-DOS is inherently a single-tasking environment, DESQview can perform this function only by imposing a substantial overhead penalty. On the prototype machine, the multitasking penalty neutralized many timing measurements obtained from the Phase II model field test.

Consequently, throughput estimates in this report are based on measurements excluding the DESQview model performance.

Telecommunications for the Phase III model were based on X.25 packet-switching protocols. The goal of the Phase III model was not to complete a comprehensive design for an operational LGN. It was to permit a realistic examination of all issues pertinent to the MODELS technical support task, matching operational functionality as closely as possible. Accordingly, several different communications media were explored with the Phase III LGN.

TRANSLATION TABLES OVERVIEW

Without a table methodology, changes between existing DLSS and equivalent EDI formats, whether major or minor, would dictate modifications in the translation software, always a major task. The objective therefore was to insulate software from the effects of evolution in transaction definitions. Software that translates literally between the two formats would run counter to such a goal. What is needed instead is software that assimilates the conversion logic expressed entirely within the translation tables. Properly constructed tables, interpreted by an executive software routine, can ensure that changes occurring in either DLSS or EDI formats will impose table-entry updates only.

Table construction for the DLSS is not straightforward. There are approximately 425 distinct DLSS transactions, falling into seven major transaction families.[1] For these, 56 equivalent EDI transactions have been designed as replacements.

Commercial suppliers of translators also approach EDI translation by using table logic. Their products, however, do not supply a simple solution to the translation issue. Locating an existing commercial translator to put the information contained in the DLSS into suitable EDI structure would solve only a small part of the translation problem. In commercial systems, translation into a standard EDI format occurs starting from a predetermined flat-file layout. The problem lies in the infeasibility of configuring DLSS data for a common flat file. Most of the effort in

[1]DLSS transaction families are MILSTRIP, Military Standard Transaction Reporting and Accounting Procedures (MILSTRAP), Military Standard Billing Procedures (MILSBILLS), Military Standard Transportation and Movement Procedures (MILSTAMP), Military Standard Petroleum System Procedures (MILSPETS), and Supply Discrepancy Report (SDR). Presently, all but SDR require translation to EDI; SDR has no equivalent DLSS electronic format.

translating DLSS information is in deciphering the encoding rules implied in using the 80-column format.

In fact, because of the complexity of the DLSS logic, the translation tables employed by the prototype LGN were divided into two sets for each direction. The first, a control set, is typical in EDI translation software; it controls the translation sequence and forms the data retrieved into the target format. A second set of tables, "EVAL-tables,"[2] defines and evaluates conditions for transformations required for an unambiguous interpretation of the DLSS data. All evaluation rules contained in the EVAL-tables are compiled into executable code during translator initialization.

Tables DLSS-to-EDI

The controlling DLSS2EDI table serves as a pointer for the translation software; it identifies data entries needed to create an EDI format. For example, the initial constituent in an EDI transaction is its ST (STart) segment, which distinguishes the transaction type. Determining the transaction type requires looking into the DLSS data and retrieving a DIC. This DIC (the simplest data conversion between DLSS and EDI) always occurs in the first three data columns and determines which EDI transaction set the translator selects from the tables. The instructions for locating and interpreting all DLSS data, most of them quite complex,[3] are invoked from the EVALDLSS table.

Tables EDI-to-DLSS

In translating from EDI to DLSS, the translation table pair operates essentially in reverse of that just discussed. That is, the EDI2DLSS control table points to the 80-column position currently being filled. The role of the EVALEDI table is to locate, evaluate, extract, and transform (as necessary) data from the correct segment and element within the EDI transaction. Of course, the length of each field is transaction-dependent.

In general, translation from the EDI format is both simpler and faster than from DLSS, because (1) the process of locating and interpreting EDI data is more

[2]The table containing functional logic used for translating from DLSS to EDI is EVALDLSS; EVALEDI contains data transformation rules for translating back to DLSS.

[3]Examples of some of the more intricate evaluations include (1) checking for the existence of specific conditions and (2) transforming data through fixed or calculated substitution.

straightforward and (2) far fewer interpretative conditions are needed to fill DLSS transaction fields from EDI data.

Test Transaction Formats

While EDI transactions exchanged in the test were constructed according to the design developed by the MODELS Functional Task, they included one additional segment. Test transactions contained a special "XXX Segment," whose contents are the entire original 80-column transaction. This test segment provided the means for comparing translated and retranslated results against the contents of the original transaction.

TRANSLATION SOFTWARE OVERVIEW

The prototype translation software was written exclusively to process translation tables. Along with unambiguously interpreting table parameters, the translation software compiles the logic of the EVAL-tables into executable code. For the operational LGN, if translation software accommodating the syntax, operators, and logic contained in these prototype tables were written, no new translation tables would need to be developed for translating between DLSS and current EDI versions.

FIELD TEST OVERVIEW

The field test of the prototype LGN was based on initial assumptions, hypotheses, and perceived alternatives. Its purpose was to develop findings, validate conclusions, and develop recommendations through actual hands-on investigation of the prototype LGN.[4] As with developing the LGN, a progressive testing sequence produces a gradual build-up of knowledge throughout the course of the test, and it optimizes the gain of information from the early stages, as a hedge against being left without information if, for some reason, the later stages are not implemented. Four technical goals were defined in advance for the MODELS prototype field test:

- Prove and clarify the MODELS concept: translate, transmit, and verify the adequacy of EDI transaction formats developed to replace DLSS fixed-length transactions for all Services and OSD, operating in a variety of data processing and telecommunications environments
- Integrate the MODELS test equipment with DAAS value-added processing: provide prototype translation capability between EDI and 80-column

[4]Highlights of the field test experience are summarized in Chapter 3.

formats, apply DAAS processing services to EDI transactions, and demonstrate coexistent processing of both transaction formats

- Demonstrate the flexibility of the EDI format for DLSS transactions: improve, extend, or add transactional capabilities
- Define a baseline MODELS operational system: develop technical specifications and recommend requirements and an approach for planning system implementation.

The field test encompassed all three phases of the prototype LGN design, translation tables and prototype translation software, local connections to host site equipment, and WAN telecommunications.

The test plan commenced with the initial standalone translator and progressed in stages to include live-data transfer between cooperating field sites. A live-data test actually replaces the existing media for transaction exchange by those of prototype test facilities. Its accomplishment depends on (1) progress made during interim test steps and (2) agreement among participants to substitute test circuits for a short duration.

The LGN prototype task – from development of a prototype LGN to testing it in the field – was intended to provide proof of concept, rather than to test an operational LGN network. Accordingly, the test plan sought to create only minimal impact on test-site operations. Such a minimal-impact philosophy offered two direct benefits. For one thing, sites would more readily agree to participate if inconvenience to them would be slight. For another, connection to heterogeneous ADP equipment is expected to pose a difficult technical challenge in an operational system, and the problems involved can be studied more clearly when isolated and not unnaturally eased by assistance from site personnel. Thus, communications between an LGN and site equipment generally were accommodated solely through the resources of the testing project. Matching communications protocols between an LGN and its host computer were utilized when possible; otherwise, the project team created the exchange mechanism. Most often, host computers simply downloaded duplicate copies of transactions before transmitting the transactions over present telecommunications facilities.

SIMULATION AND COST MODELS

We chose two additional investigative tools – a simulation or performance model and a cost model – to gain insight into issues falling outside the province of the field test. The requirements for the simulation readily fall into areas of (1) functional performance and (2) cost assessment. Together, the performance and cost models provided answers concerning cost-effective levels of distribution of EDI transaction processing. For the performance model, there were three goals:

- *Function distribution*: Determine the best level of distribution for MODELS functions. Which functions should be distributed and which centralized?
- *Communications impact*: Determine the communications requirements to support a network of interconnected LGNs. Characterize the variance in traffic-load demand depending on functional distribution.
- *Network alternatives*: For given levels of distribution and requirements, determine available network alternatives.

Building on results from the simulation, the cost model evaluated cost impacts of various alternatives:

- *LGN size*: On the basis of current volumes of logistics transactions, determine appropriate LGN configurations and cost.
- *LGN number*: Determine the optimal number of LGNs required to meet expected needs.
- *Communications costs*: For specific network configurations and traffic loads, determine the expected costs. Conduct sensitivity analyses of cost versus level of distribution and across various network alternatives.

CHAPTER 3
PROTOTYPE TEST

Generally speaking, the field test was conducted in a manner consistent with the prototype test plan. Appendix A points out deviations by comparing actual testing procedures with excerpts from the test plan.[1]

The test yielded essential data that assisted with (1) designing and developing the prototype LGN, (2) developing and correcting translation tables, (3) identifying and locating the source of anomalies in transaction use, and (4) developing a baseline performance specification for an interconnected network of operational LGNs.

Test results from the field test sites were documented formally for a period of a few weeks. The test data are published in the third volume of this report, *Logistics Gateway Node Technical Specification.*

THE FIELD TEST EXPERIENCE

All test sites were equipped with nearly identical prototype hardware and software. While actual operation required individual site tuning, the general information flow for the host log-on process and download script described in the next paragraphs applies to all the sites in the test. Any significant exceptions are noted.

Download Process

Data are transferred from host to LGN via a polling operation occurring within a "download window." During each poll, the local interface module invokes the downloading process, which in turn calls one or more site-dependent programs to perform the following functions:

- Establish a connection with the host (if the connection is not permanent)
- Log on to the host
- Download all available designated files

[1]A prototype test plan, intended as a guideline for prototype testing procedures, was published in a November 1988 LMI Temporary Report, *MODELS Test Plan*, DL702TR1.

- Log off the host
- Update the file that designates the currently active download window
- Disconnect from the host (if the connection is not permanent).

The main module waits a designated period for a message indicating that the download attempt has finished. If this message is not received within the time allotted, the software assumes that a nonrecoverable error has occurred during the host session, the timer is reset, and processing continues.

Host downloads may take a relatively long time. In the prototype LGN, to preclude interference between downloading actions, download windows were scheduled far enough apart to ensure that any in progress were completed before the next window began.

Script Program Processing

The following describes the processing flow of the script program; it is applicable to almost all of the sites in the Phase III prototype test (the complete script is presented as an integral part of the prototype LGN design in Volume III of this report):

- After reading the download window parameters and performing other program initializations, the script attempts to log the LGN on to its host. If the attempt is successful, an appropriate return code is passed; otherwise, the script logs off the host, returning control to the download script batch process.
- Once the LGN is connected to the host, the script downloads a host file. During the prototype test, the precise software and methodology used varied from site to site; whenever possible, file-transfer software having cooperative host and LGN components was used.
- After the download, the script checks for the file and a return code. Their presence does not guarantee that the download has been flawless – only that there is a result.
- A message disclosing the name of the downloaded file is sent to the LAN dequeueing module, and the local interface module resets the timer.
- All pertinent events[2] are logged.

[2]These include timer and parameter values, the start and end time of the download script process, and the size and create-date of the downloaded file.

As noted, these steps portray the generalized procedure. In practice, all sites required at least minor modifications, and in some instances notably diverse host environments were encountered. Appendix B characterizes the variability experienced during the development and implementation of the local interface portion of the LGN at two sites: Naval Supply Center, Norfolk, Virginia, and Troop Support Command, St. Louis, Missouri.

SIMULATION AND COST MODELS

The simulation called for a high-level tool incorporating existing modeling structures adaptable to the MODELS scenario. In general, the requirement was to create the major hardware and software elements of an operational transaction-exchange system. Major programming effort was to be avoided where possible. What was needed was a simulation package suitable for gathering basic statistics, rather than one providing for extensive modeling of fundamental data structures.

Network II.5, a Simscript-based software program offered by CACI and closely fitting the requirements, was selected for the simulator. It contains basic programming elements[3] supporting hardware and software simulation of a computer network. Further, it provides measurements of hardware utilization, software execution, and resource conflicts for a user-specified computer system description. The Network II.5 design permits evaluation of (1) a system's ability to handle a workload and (2) performance of alternative system configurations. Target system configurations can be created through a menu-driven interface and processed by a built-in statistics package.

Developed separately from the Network II.5 simulator, the cost model employed data provided by the simulation model. The cost model design made it possible to derive both fixed and variable costs for each of three simulation scenarios (see below). The simple cost model was implemented via a Quattro Pro spreadsheet. Scenario configuration costs were viewed over a full life cycle, assumed to be 10 years.

[3]Network II.5 provides for three types of hardware devices: (1) processing elements, (2) transfer devices (for transmitting data), and (3) storage devices. Software components use the hardware devices to accomplish various tasks. There are four types of software components: (1) modules, which employ hardware components to process, read, write, and transmit information; (2) instruction mixes and macro instructions called by modules to handle repetitive tasks; and (3) files that reside on and can be passed through the network to (4) storage devices. All of the devices, both hardware and software, are discretely created and specified by parameters.

Input Values and Processing Compromises

In simulation, compromises are necessary. For example, the threshold for deployment of LGNs was set to exclude those sites with daily volumes below an arbitrary threshold. All lower volume sites were assigned to transmit collectively to a CLGN. Also, as a measure to simplify matters without compromising the ability to draw reliable conclusions about LGN configurations, actual file exit times onto AUTODIN were ignored for all sites.

Three disparate simulation scenarios measured the effect of various levels of LGN distribution: (1) a centralized approach, with all transactions passing through a CLGN located at a DAASO site; (2) a decentralized configuration, with all transactions directly transmitted LGN-to-LGN; and (3) a partially decentralized arrangement under which, for high-volume sites with on-site LGNs, certain transactions identified as candidates for direct transmission between LGNs bypass the CLGN.

Projecting today's traffic patterns onto these scenarios (1) establishes bounds for operating network variances, (2) permits realistic inferences to be made, and (3) provides supportable rationale for introducing simulator results into the companion cost model.

In the cost model, three complete hardware/software package configurations of the AT&T 3B2,[4] corresponding to high-, mid-, and low-range capabilities, were evaluated. No equipment charges were entered for non-LGN sites; it was assumed that those sites already possessed basic telecommunications and processing capabilities.

Volume Threshold

Early in the task, a simplifying decision was made that only sites whose volume exceeded 10,000 transactions per day would be discretely modeled as high-volume sites. This compromise was made in the light of three considerations.

- The simulation load must be one that the simulator workstation can handle in a reasonable time (an overnight run, for example). Adding sites to the network rapidly increases the run time for a simulation.

[4]The 3B2 model is a UNIX-based microprocessor currently available to the Government through contract with AT&T.

- There are only a relative handful of large-volume sites; thereafter, volumes rapidly drop; a cutoff point of 10,000 daily transactions eliminates all but 149 sites to be modeled.
- On the basis of projections from a 16-node test case, it appeared that a full day of traffic for a network of 149 nodes could be simulated in an overnight run. Such a 149-node benchmark includes all major logistical activities and representatively samples typical operational military sites.

Time Scales

For the model, the ideal simulation time would cover network operation over the course of a single day. Since a workday should show any effects of hourly surges in traffic, there seemed to be no advantage in modeling greater periods. Except for periodic yearly fluctuations, for which reliable data were not available, most desired effects could be observed from using a 1-day timeframe.

The sites were modeled on worldwide time zones and operated in a staggered manner. However, since the major sites were all in the Continental United States and were separated by few if any time zones, the simulated operation was compressed into an 8-hour window. Comprehensive detailing of time would be difficult to implement and might actually result in lessening the average stress on the network and on individual sites. While lacking absolute precision, the 8-hour simplification did normalize results across the sites.

TREATMENT OF DAAS

As with the actual logistics network, the DAAS site is a critical component in the modeling process. In the centralized and partially decentralized scenarios, DAAS carried the bulk of the traffic. In all three scenarios, DAAS served as the interface for non-LGN traffic and as an end point for image transactions.

DAASO currently is undertaking significant modernization efforts. A review of some of the likely improvements showed the difficulty of projecting the impact of new DAAS capabilities within the context of the network simulation. Moreover, any advantages accuring from enhanced results in the simulation would be outweighed by the disadvantages imposed by the complexity of detailed modeling of present or future DAAS capabilites. Consequently, for purposes of the simulation study, DAAS was simply modeled as a routing node with a built-in, but variable delay.[5] In the simulation, transactions sent to DAAS received the following simulated processing: (1) the header is read, (2) DAAS places the messages in a queue, and (3) after a delay in simulation time, DAAS transmits them to the final destination.

LGN Site Design

Two of the AT&T 3B2 configurations reflecting candidate designs promulgated by DAAS and by Lawrence Livermore National Laboratory were priced as complete hardware and software packages. Both were entered in the performance model and in the cost model. Maintenance charges for the configurations, available from AT&T, were included for each year.

LGN external interfaces were quite complex, because of the significant variety among LGN-host connections, as verified by the field test. Modernization activity at many of the prototype test sites indicates current or planned facility-wide networking. The simulation model supposed an LGN-to-host connection across an Ethernet-based LAN. Connections to the DDN WAN were presumed to be dedicated, with transfer rates of either 56 or 9.6 kilobits per second (Kbps), depending on traffic volume.

Presently, transactions are batched periodically at the test sites and transferred (often manually) to the AUTODIN facility. If an LGN has access to its host and is activated in real time, small packets of transactions could be transmitted continuously over DDN. However, for the simulation, it was assumed that the present batch generation of transactions in the host system would not change.

[5]Simulation rules for DAAS include route packets by header (depending on scenario), add a constant processing delay, and perform translation for non-EDI and non-LGN sites. The delay in transmission time represents overall DAAS processing and is not transaction- or destination-specific.

CHAPTER 4

CONCLUSIONS

Because of the wide range of issues investigated in the MODELS technical support task, there are several conclusions and results to list. These are organized topically and discussed without ranking.

PROTOTYPE CONSTRUCTION AND OPERATION

Complete details of the prototype LGN are published separately in the second volume of this report, *Logistics Gateway Node Prototype Construction and Operation.* The following section describes the results and conclusions derived from constructing and operating the prototype LGN as part of the field test.

LGN FUNCTIONALITY

The size – and therefore the cost – of the LGN machine depend directly on the functionality required. Minimization of LGN cost is served by confining LGN tasks to the least number of required functions. The cardinal role of an LGN – those tasks that it must perform – can be divided into six categories:

- *Exchange of transactions with host processor*: This exchange includes all procedures and protocols necessary to permit downloading of data from a host to its LGN and retrieval by a host of data logged at its LGN. The job also entails creating and maintaining message and error logs to be exchanged with a host processor.

- *Translation between DLSS and EDI formats*: In addition to table-driven translation, this task encompasses selection of transactions requiring translation. As an example, host-generated EDI transactions would be culled to bypass translation and would be queued directly for transmission. Also, incoming EDI transactions for which a host is EDI-capable require no translation. The translation chore involves all prescribed edits for outbound transactions, including any host-generated transactions already in EDI format.

- *Compression/expansion of transactions before/after transmission*: In contrast to transactions handled using past logistics telecommunications practices, EDI transactions will be compressed prior to transmission over the WAN. Hence, they must be expanded upon receipt from the WAN.

- *Public key encryption (PKE) encoding of transactions for transmission*: Logistics transactions presently are not classified. However, in aggregate numbers, logistics data are considered sensitive. Encryption before transmission and decoding upon receipt may be required to offer adequate security protection for EDI transaction exchange.
- *Exchange of EDI-formatted transactions across a WAN*: This task entails more than merely matching end-to-end protocols for packet-switched transmissions. For example, it includes routing outbound transactions to an intended LGN or intermediate CLGN by address-table look up. Also, copies of transmitted data need to be retained at the LGN until their receipt is acknowledged, and selected file copies need to be forwarded to the CLGN.
- *Operational control*: This is an administrative task for managing all updating of LGN translation tables, address tables, and timing and control parameters through commands remotely issued at a CLGN.

Multitasking

To support the LGN's basic, minimum functionality, a multitasking faculty is required. A single-tasking machine can perform tasks only in strict sequence, awaiting completion of prior tasks before starting new ones. Without multitasking, functional activities in the LGN cannot be prioritized.

Data Compression

Converting to EDI format increases transaction size. Unless modified, the EDI transactions are double the size of equivalent DLSS transactions.

The compression software used in the field test is a standard set of compression algorithms released by PKware, Inc. Using them reduced EDI test transactions to approximately one-fourth their previous size. Assuming equivalent compression techniques for an operational network, EDI formats can be exchanged at one-half[1] the size of current, uncompressed fixed-length (DLSS) data.

Since transmission costs correlate directly to the amount of information conveyed, EDI transactions should be minimized in size before they are released to

[1]That is, since uncompressed EDI transactions are twice as long as their DLSS equivalents, reducing them by three-quarters gives an EDI result one-half the size of the original, uncompressed DLSS transaction.

the WAN transmission media. Data compression is mandatory to prevent cost escalation.

Transaction Editing

Most data fields in a transaction are edited automatically during translation, particularly with respect to data typing. For example, if a field entry is to be translated as a date, it must contain date-type data. Any transaction containing incomprehensible data should be rejected at the LGN, if only to limit wasteful transmission costs. Any exceptions to the standard practice of rejecting erroneous transactions should require justification on a case-by-case basis.

Rejected transactions can be returned in their original format with a narrative message to the host computer. The LGN editing process can assign rejection codes for errors detected and generate a corresponding narrative message for each code explaining the cause for rejection. Prudently constructed narrative messages can help host facilities personnel correct faults before transactions are reintroduced into the logistics transaction pipeline. Again, exceptions to rejection should be justified on a case-by-case basis.

The editing of transactions generated in EDI format introduces an interesting dilemma. If all EDI transactions created within a host were machine-generated, editing should not be required. Is it not wiser to correct the host software than the transaction? However, some EDI transactions entering the LGN may be produced manually; properly, these would benefit from editing in the LGN.

EDI transactions received at an LGN inbound from the WAN, regardless of origin, need no editing; that will have occurred outbound at the initiating LGN. Basic edits[2] consistent with a small-machine strategy for the LGN can supplement the functional editing that a transaction receives under the DAAS processing rules.

Transaction Routing

Some transaction sets, [Military Standard Contract Administration Procedures (MILSCAP), for example] contain no from/to indicators within the data. For these, the addressee should be designated within the transmitted EDI envelope. To

[2]The LGN can readily establish the presence of data, but without the capacity of a large central processing facility, in many situations it could not determine the validity of information entered in the data field.

eliminate transaction-dependent addressing schema, serious consideration should be given to using the EDI envelope for addressing every EDI transaction.

In the present telecommunications system, DLSS-formatted logistics transactions are carried over AUTODIN, are addressed to one of the DAAS value-added processors, and are routed by DAAS to their ultimate destinations. Technically, one LGN can transmit EDI transactions over DDN directly to another by address-table look-up. In practice, address-table entries initially will contain DAAS-site CLGN addresses only. For as long as is necessary, DAAS processors can continue to receive transactions from the WAN via their CLGN and can perform destination routing.

The calculations for determining final destinations for EDI transactions should remain a central function of DAASO. Replication at remote sites of equipment and data necessary to distribute routing processing is not consistent with a strategy for minimizing LGN hardware cost. Furthermore, for as long as the DAASO mission includes special functional requirements limiting the number of transactions eligible for direct routing (those requirements include Defense European and Pacific Redistribution Authority, Foreign Military Sales, Logistics Information Processing System, 90-day and 1-year transaction history recovery, and transaction image processing), only a few transactions are candidates for direct routing between LGNs. If the list of directly addressable transactions grows significantly and if DAASO's mission responsibilities change to permit LGN-to-LGN transaction exchange, direct addressing could be accommodated in a technically simple manner. The following steps illustrate an approach to direct addressing that is viable and is consistent with a small-machine LGN strategy:

- For every outbound transaction, the LGN scans its address table in search of a network address entry. At LGN start up, the table contains only CLGN addresses, so routing for all transactions defaults to a CLGN. Thus, the final routing for every transaction occurs initially at DAAS.
- At the DAAS CLGN, transactions matching those types contained in a master list of directly addressable candidates are earmarked.
- After DAAS has calculated the destination address for the earmarked transactions, the CLGN returns to the originating LGN the address ascertained, along with rules (i.e., conditions) qualifying its application.

- Upon receiving a network address and rules from the CLGN, the LGN updates its address table. Note that each LGN develops its own unique address table through this process.
- Thereafter, when a transaction entering the LGN obeys all the rules and has from/to entries matching the address-table entries, the LGN uses its address table to route the transaction directly.
- Of course, we need a means for rescinding LGN address-table entries. All that is required is a simple delete command broadcast to all LGNs. Upon receiving such a command, all LGNs with matching table entries simply eliminate the cited address and rules from their tables.

REDUCTION OF DAAS VULNERABILITY

A critical telecommunications hub, DAAS is perceived as vulnerable to natural and man-made catastrophe, for which no simple correction has been developed. This elementary direct-addressing scheme offers a solution for contingency operation in the event of central processing facilities outage. In that event, requisitions and other important traffic would be routed directly, according to entries in a contingency address table. Such a table could be created and retained on the hard disk of every LGN for emergency back-up.

After all logistics activities have completed the transition to processing directly in EDI-formatted transactions, the need to translate between DLSS and EDI will of course end. Consequently, the long-term role for a system of interconnected LGNs may be to provide a necessary contingency operation.

TRANSLATION TABLE CONSTRUCTION

Because making the transition from current DLSS formats to exclusive use of EDI likely will occur over a protracted period, there will be a lingering need to convert between these formats. Moreover, during the transition and afterward, EDI transaction definitions will evolve[3] toward a more enriched functionality. Table-driven software offers a flexible, long-term means for supporting the exchange of transactions between activities that are at differing stages of EDI implementation.

As previously noted, while most commercial EDI translation processes use tables and translate to EDI from an unambiguous fixed-format flat file, DLSS

[3]At the close of the prototype task, version 1 translation tables were developed and tested. These are presented in their entirety in Volume III of this report.

translations require more than just moving data into and out of the EDI format. Unraveling the contents of fixed-length formats having multiple definitions represents the bulk of the DLSS conversion requirement. Interpreting DLSS data requires a mapping between data elements of the DLSS and EDI formats that is unavailable commercially. Consequently, for both DLSS-to-EDI and EDI-to-DLSS applications, the prototype test experience suggests using two tables in each direction of translation. One set of tables controls the sequence by requesting, in turn, data elements needed to satisfy the new format, while the second set (EVAL-tables) locates and evaluates each requested datum from the original DLSS or EDI format. Compiling the EVAL-tables, which employ a high-level pseudo-language, into the translator program as macro-routines significantly speeds up the translation process.

DLSS transactions requiring multiple images [e.g., Military Standard Billing System (MILSBILLS), MILSCAP, and Military Standard Transportation and Movement Procedures (MILSTAMP)] may entail lengthy sequences of 80-position data, all pertaining to a single transaction, and may be presented for translation in virtual random order. To keep translation tables manageable and maintainable while accommodating DLSS conventions, each fixed-length image must be autonomous. Organizing linked images for presentation to the translator in a usable succession requires some housekeeping tasks. These can occur in a pre-process outside the tables and translation software. Examples of such tasks include DIC selection, sorting and ordering of multiple transaction images, and recognition of transactions not requiring translation.

Translation tables presently deal with transactions whose contents fit 80 columns of data. Even before variable-length DLSS-equivalent transactions will have been employed in the field, baseline EDI transactions definitions will already have been augmented. These enhancements contain extra-DLSS data that cannot be translated into a DLSS format because no DLSS equivalent exists. As a practical measure for enhanced EDI transactions arriving at an LGN whose host system is not equipped for EDI, the LGN can (1) translate the standard transaction's core elements to DLSS and (2) hold the entire EDI version for a time (say, 3 days). Within the specified period, host-site personnel can pick up such transactions from the LGN for manual processing.

TRANSLATION SOFTWARE

A benefit of consolidating all translation logic into tables is that the translation software can be kept simple and maintainable and, once written, need not be further subject to modification. Software whose exclusive task is to processes tables – such as those developed in the prototype study – is isolated from transaction-specific logic. Thus, as often as transactions are modified, (1) only the table logic need reflect any changes and (2) after re-initializing (compiling), the software will continue to operate without adjustment. Accordingly, a table-driven translator performs just three basic roles:

- It interprets the expression syntax and logical operators employed in the translation tables.
- It selects the correct master table for each transaction and operates under its control.
- It compiles and executes pseudo-language routines from the EVAL-tables.

Translating speed is critical, because high concentrations of logistics transactions are exchanged among many activities. Commercial EDI translation software (from flat file to EDI) typically converts between two and six transactions per second. Some DoD logistics sites routinely process as many as 450,000 transactions a day; that is over five transactions per second. For those sites, any surge could result in an overload in the flow of outbound and incoming transactions, given a six-per-second translation rate. Assuming a translation capability in the range of six transactions per second, for an operational system, a safety margin is required. It could be gained by deploying a second or a third translation processor at high-volume sites.

Presently, at the central nodes, DAASO handles about 80 million transactions a month. Imputing an even distribution to that volume exceeds 30 transactions per second, 24 hours a day, every day. Realistically, the traffic is neither evenly distributed between the two DAAS sites, nor does it arrive smoothly, but in spurts. Adding a 50 percent safety margin for the central DAAS sites, the minimum translation rate required is approximately 45 transactions per second.

PROTOTYPE FIELD TEST

One aim of the field test was to provide practical insight into operational difficulties between the LGNs and their interconnecting network. Only through hands-on experience can the differences between expected and realized results be highlighted. Several conclusions regarding operational options can be drawn from the field-test experience.

Host Interconnection

From the outset, the greatest challenge for the field test appeared to be that involved in communicating between a prototype LGN and its test-site host. The challenge proved to be even greater than expected.

To benefit from a connected LGN, a host must communicate with it regularly, quickly, and reliably. The principal lesson learned from the field test was that the interconnection between LGN and host can be a complex obstacle to successful performance. At a few of the test sites, acceptable operational interconnection was never established. Before an operational system can be practicable, the LGN-to-host interface problem must be simplified. Fortunately, simplification is not difficult. But the host processor must be required to communicate with its LGN, choosing one from the more prevalent access protocols listed in Volume III of this report.

Multitasking Capability

Multitasking capabilities are a practical necessity in an operational LGN network. However, the prototype hardware, which was selected for its overall attributes, employs the single-tasking MS-DOS environment. To test a multitasking capability, the linchpin installed in the test-site Phase III LGNs was DESQview software. DESQview emulates a multitasking environment, but at a cost. Its throughput is significantly degraded in comparison to that of a single-thread version of the LGN. Before multitasking was introduced, single-image transactions could be translated from DLSS to EDI at a rate faster than 16 per second, and the rate exceeded 20 per second for EDI to DLSS. Saddling the prototype LGN with DESQview's overhead reduced throughput to about 6 EDI transactions per second. Were the LGN to be implemented on a platform with a multitasking overhead comparable to that of the prototype hardware, sustaining minimal throughput requirements at many higher volume logistics activities would require larger capacity machines with internal clock rates greater than 20 Mhz.

Telecommunication Alternatives

As part of the field test, a comparison between capabilities of DDN telecommunications with those of alternative commercial packet-switching networks was planned. Unfortunately, DDN was not available for testing at any site in the test. Even so, clear lessons were learned regarding telecommunication alternatives.

Technically, packet-switching networks perform nearly identically. However, that is not to suggest that there are no differences among them. Large variances occur in individual pricing schedules, availability and reliability of service, and capacity and speeds. For example, the network chosen for the prototype test offered an attractive operating cost but proved often to be unavailable for service. Unquestionably, an erratic performance such as that experienced in the test could not stand up to the rigors imposed by an operational network of LGNs. Consequently, the choice of telecommunications for an operational network is critical and must trade off such parameters as cost – which indeed is important – against availability for service – which is paramount.

Validation of Transaction Design

A benefit expected from testing in the field, as contrasted with processing "canned" transaction data in a laboratory environment, was detection of deviations in site-generated transaction usage. The test plan called for transactions to be selected first on the basis of translation table capability; then, as the more complicated table logic was completed, the selection criteria would broaden. The plan worked well. Actual site-generated data provided the needed variability for effective debugging of the table logic. Moreover, approximately 60 anomalies in DLSS transaction usage were uncovered, some necessitating modification to the EDI transaction definitions.

The field test was terminated earlier than planned. As a result, those transaction groups that were tested last were not as fully explored for anomalies as were the first. As a matter of interest, the unexpected curtailment of the field test proved the wisdom of having decided upon the staged, step-wise approach embodied in the test plan.

SIMULATION AND COST MODEL

Simulation Model Parameters

In ascertaining an economically appropriate level of processing distribution for a network of interconnected LGNs, extremely complex issues are at stake. Just as with the field test, understanding of the simulation issues increased as the project proceeded. An iterative approach to developing the performance model assisted in obtaining useful results.

A general operating scenario for the LGN network was devised for the performance model. From that starting point, variations in connectivity, the number and size of LGNs, and other parameters were refined until a valid architecture resulted. The performance and cost models were maintained as separate entities in the simulation task. That separation reduced some of the inherent complexity and, at the same time, provided flexibility as changes were needed.

As planned, the performance model evolved around three distinct scenarios:

- A centralized approach, with all transactions passing through a CLGN located at a DAASO site
- A decentralized configuration, with all transactions directly transmitted LGN-to-LGN
- Limited centralization, with roughly 149 high-volume activities possessing on-site LGNs partially bypassing a central node via direct LGN-to-LGN transmission.

The most significant detail in the simulation was in modeling the LGN itself. Specific processing modules and their functions were identified from the design study. The table-driven translation software developed for the prototype LGN provided large samples of transactions and the times required for their translation, from which specific benchmarks were established. From these data, significant functions to be implemented in the simulation were selected and processing times were projected.

Decisions regarding LGN external interfaces were more complex. The field test revealed a significant variety in the connections between LGNs and their hosts. Modernization activity at many of the prototype test sites indicated current or planned facility-wide networking, usually Ethernet-based. On the basis of this knowledge, the simulation model supposed an LGN-to-host connection across an Ethernet-based LAN. Connections to the WAN were presumed to be dedicated, with transfer rates of either 56 Kbps or 9.6 Kbps, depending on traffic volume.

Detailed modeling of DDN was deemed not critical to the task, although DDN performance was investigated. Presently, while DDN is operational at many sites, it is often not functional at the required levels. Further, the field test showed that gaining access to DDN is frequently difficult, and at a facility without DDN, acquiring it is a lengthy process. Still, a number of logistics activity sites either have or are expecting to have DDN shortly. Consequently, the performance model was built incorporating specified DDN performance parameters, under the assumption that LGN sites would have DDN access at the required performance level. The model would not attempt to re-create DDN nodal structures, since it is not clear which LGNs would be assigned to any given node or how heavily the nodes would be loaded with non-MODELS traffic. Further, DDN itself is undergoing a number of technical changes to improve its operational efficiency. Therefore, DDN exists in the simulator as a single discrete entity for routing EDI transactions, and the model embeds the specified DDN performance into the communications link parameters.

Cost Model Parameters

Communications charges for all scenarios, based on current transaction-exchange workloads, were derived from the output of the performance model. For example, the performance model determined the number of transmission packets for a day's worth of traffic. That quantity was annualized to kilo-packets and entered into the cost model. Cost model parameters also took into account packet growth rate, nominally set to 5 percent per year. Communications workload factors included dedicated line sizes; larger sites use 56 Kbps circuits, while others operate at 9.6 Kbps. These considerations all reflect differences in expense and played a part in the cost evaluation. The cost model included both fixed and variable costs for each simulation scenario.

Cost Model Results

Viewing scenario configuration costs over a full 10-year life cycle gave a representative balance of initial procurement costs and variable recurring charges.

Appendix C contrasts costs derived by using various telecommunications options in the cost model. Results from the cost model are summarized as follows:

- The packet cost of transaction exchange dwarfs all other operational costs, including equipment cost and maintenance.
- Direct routing offers minor cost savings (based on maximum exploitation of the present limited number of candidates for direct routing).
- A single dedicated AT&T 3B2 (base configuration) could handle traffic volumes for most sites.
- From a cost perspective only, the most effective scenario involves using commercial dedicated and dial-in telephone lines between remote LGNs and the CLGNs (i.e., no packet-switched transmissions). However, other considerations support the concept of using packet-switching networks or other arrangements that permit direct routing and contingency operations.

CHAPTER 5

RECOMMENDATIONS

The following are recommendations derived from conducting the prototype LGN test. Adopting these recommendations can bring about significant improvements over current operations: (1) DoD trading partners will be able to phase in their implementation of EDI transactions without waiting for readiness among all Services and Agencies, (2) the acknowledged vulnerability of the present two-site telecommunications hub can be alleviated, and (3) the functional capability called for in OSD directives can be met through a low-cost investment for each logistics site.

CLGN PROVISIONS

- Develop internal processing capability for EDI-formatted transactions within the DAAS modules, eliminating any requirement for retranslation of inbound transactions from EDI to DLSS. Provide for translation of inbound DLSS-formatted transactions and outbound (EDI to DLSS) transactions to support DLSS-only (non-EDI) sites without a local LGN.

  Alternative: Continue processing present internal format at DAASO, translating from EDI. (Not recommended, because throughput at the central hub could deteriorate as a result of the added translation load, unless processing power at DAASO were significantly increased.)

- Where telecommunications other than the primary WAN are used (for example, between DoD logistics activities and commercial contractors), provide gateway access to secondary networks at the CLGN only.

  Alternative: Provide true internetwork gateway services at all LGNs. (Not recommended, because rendering internetwork service at a local LGN will significantly increase variability among LGNs, increasing both their cost and the difficulty of their maintenance and configuration management.)

- Develop and implement a method for CLGN-controlled address tables (consistent with that described in Chapter 4) to regulate logistics data transmission between LGNs and their assigned CLGN and from LGN to

LGN. Develop a means to create, maintain, and invoke contingency[1] address tables at every LGN for emergency use in case of CLGN outage.

Alternative: Continue to provide only centralized routing and processing of EDI transactions at DAASO. (Not recommended, because (1) no contingency operation would be provided and (2) increases in regular ongoing logistics traffic will necessitate expansion of central facility capacity requirements.)

- Develop a command instruction set from CLGN to LGN and from the on-site manager, to control each of the following:
  - Replacing entire translation table
  - Deleting entire translation table
  - Deleting translation table line entry
  - Inserting translation table line entry
  - Replacing entire address table
  - Deleting entire address table
  - Deleting address table line entry
  - Inserting address table line entry
  - Replacing entire host EDI-capability table
  - Deleting entire host EDI-capability table
  - Deleting host EDI-capability table line entry
  - Inserting host EDI-capability table line entry
  - Replacing entire parameter table
  - Deleting entire parameter table
  - Deleting single parameter table line entry
  - Inserting single parameter table line entry
  - Suspending LGN operation

[1]Providing a cost-effective means for contingency operation offers a potential long-term role for the MODELS LGN. After the transition is completed, when all activities are EDI-capable, there will be no need for translation between DLSS and EDI formats. Moreover, management of differing levels (versions) of EDI usage among DoD trading partners falls squarely within the capability of commercial translation software and services.

- Restarting LGN operation
- Requesting LGN logs to be sent to CLGN.

Alternative: Provide for autonomous LGN operation. (Not recommended, because (1) synchronization among a large number of participants would be virtually impossible to maintain with individual control of table and software updates, operational controls, etc., and (2) the objective of distribution would be to provide a rational dispersion of value-added functionality for improved efficiency and flexibility, not mere decentralization of control.)

DEPLOYED LGN PROVISIONS

- Deploy small, identical multitasking LGNs at major operational field sites whose characteristics meet, insofar as is practical, the specification developed from the prototype test and documented in Volume III of this report.

  Alternative: Have DAASO continue as a central hub processing logistics transactions, with no deployed field site LGNs. Because translation between DLSS and EDI is an interim transitional requirement, provide transaction translation via the DAAS CLGN between DLSS-only sites and EDI-capable sites. (Not recommended, because (1) throughput at the central hub could deteriorate as a result of the added translation load, unless processing power at DAASO were significantly increased, and (2) continued total reliance on a central processing facility offers no contingency protection.)

- Develop a standard method to initialize, and software to utilize, an LGN parameter table containing operating values that fine-tune host and LGN interaction. These parameters would govern matters such as (1) preference between real-time and batch processing, (2) daily schedules for uploading and downloading batch transactions, (3) number of re-tries and time-out period before disconnecting between host and LGN, (4) list of EDI transactions of which the host is capable, and (5) list of Service-unique transactions that the LGN should pass without translation and, (probably) without editing.

  Alternative: None, if remote LGNs are deployed.

- When expanded capability beyond the standard LGN configuration is required at a site, do not increase the individual LGN's capacity; rather, increase the number of LGNs at the site.

- Develop a standard software suite for interconnecting multiple LGNs at a site and for distributing workloads among them. This control software could be resident in every LGN and invoked by parameter table if additional LGN units are installed at a site.

Alternative: Define LGNs with differing capacities to accommodate site-dependent workloads. (Not recommended, because LGNs of different configurations will increase both equipment purchase price and long-term maintenance costs.

- Limit modes for communication between a host and its LGN to the alternatives[2] listed in the LGN specification. As noted, the LGN-to-host interface problem must be simplified in a practical operational system. Since the LGN configuration is standard but host configurations are not, the solution for the interconnection technical trap – amply exhibited in the field test – is to require the host to interface with the LGN standard.

  Alternative: Provide site-specific interconnection between the host and LGN. (Not recommended, because doing so would entail differing LGN configurations, increasing equipment purchase price, and adding to long-term maintenance costs.)

TRANSACTION EDITING PROVISIONS

- Edit outbound DLSS transactions as a by-product of the translation process. Except for a few transactions,[3] return all failing the translation edit to the host. Return standard diagnostic messages with them, explaining the cause for their rejection. Require correction of host computer software to eliminate future recurrence of erroneous DLSS transactions. Continue to provide customary functional editing at the centralized DAAS facilities.

- Perform pass/fail editing on outbound EDI transactions generated at a host. Return all failed transactions (except as noted) to the host. Correct host computer software to eliminate future recurrence of erroneous EDI transactions.

- Perform no LGN editing on transactions inbound from the WAN, since they will have been edited when outbound at the originating LGN. (An exception is transactions received at a CLGN from sites without an LGN. For these small-volume sites, the CLGN performs originating-LGN duties, and customary functional editing is provided by the DAAS processor.)

- Write failed transactions, both EDI and DLSS, to an error log at the local LGN for return to the host and identify incorrect data elements. (Not only does rejection at the LGN save time in turning around faulty transactions, it

[2]The recommended interfaces are: CSMA/CD 8802/3, Token Bus ISO 8802/2, Token Ring ISO 8802/5, IBM 3270, and asynchronous communications such as Kermit or XMODEM.

[3]Selected transactions related to logistics transportation processing may be excepted. Transactions containing "paperwork" that are needed at the depot and whose timely arrival may be more critical than the integrity of their contents should be sent immediately, regardless of their error content.

contributes to reducing the highest-cost element of the transaction exchange process: data transmission.)

- Provide for retaining a previous-day log at each LGN. As an additional final edit, check outbound transactions against the history log to eliminate duplicate transmissions.

  Alternative 1: Perform a full functional edit at the LGN. This alternative is attractive because the expense of outgoing and return transmission of faulty transactions could be virtually eliminated (data transmission, as noted, is by far the most costly element in the transaction exchange process). (Not recommended, because (1) most faulty transactions can be eliminated via the recommended limited edits and (2) the cost of increasing LGN capacity to allow full functional editing would outweigh the gain from eliminating the remaining, relatively few erroneous transactions.)

  Alternative 2: Perform no edits at the LGN (except as a translation by-product). An enticing argument for this alternative favors correcting site software – once and for all – to eliminate proliferation of computer-generated errors. (Not recommended, because control over the editing role among a large number of participants would be difficult to synchronize and maintain.)

DATA TRANSMISSION PROVISIONS

- Use standard, commercially available software to compress outbound EDI transactions to no more than one-quarter of their uncompressed size. Expand compressed data at the receiving LGN through use of matching off-the-shelf software.

  Alternative: Transmit EDI transactions individually from LGN to LGN on a continuous basis, without bundling and without compression. Immediate transmission of single EDI transactions furnishes the only argument for by-passing data compression. (Not recommended, because (1) for virtually all logistics traffic presently exchanged, direct LGN-to-LGN routing is not yet practicable, (2) bundling of transactions bound for a common destination imposes little processing delay in today's operational environment, and (3) since telecommunication is the largest cost factor in the exchange process, data compression is an economic requirement.)

- Use standard, commercially available PKE software to protect outbound transmitted data. Protection of transmitted logistics information is necessary because aggregated logistics information has a security sensitivity greater than that of its component parts. But since logistics data at a field site are neither classified nor sensitive, provide only ordinary password access protection at LGNs.

Alternative: There is no alternative to encryption, simply because – taken all together – large amounts of logistics data are highly sensitive. Fortunately (from a technical perspective), logistics transactions remain unclassified, and PKE protection should suffice.

- Use DDN where possible for interconnecting remote LGNs. If an alternative WAN must be employed, select it, first, on the basis of availability for service and, second, on the basis of cost. Connect LGNs to the WAN either via dedicated or dial-in service, depending on traffic volume requirements. When available, and as required, employ X.400 protocol across the WAN.

  Alternative: Use only dedicated and dial-in service between the CLGN and remote LGNs. (Not recommended – even though the cost model indicates that such an arrangement is attractive – because of the impossibility of implementating direct LGN-to-LGN transmission and because it does not solve the problem of contingency operation.)

- For enhanced[4] EDI transactions arriving at an LGN whose host system is not equipped for EDI, the LGN should (1) translate all core elements of the standard transaction to DLSS and (2) hold the entire EDI transaction for 3 days to permit manual retrieval for subsequent processing.

- To eliminate transaction-dependent addressing schemes, standardize on employing the EDI envelope for addressing all EDI transactions.

  Alternative: Continue to use information embedded in the transaction for routing. (Not recommended, because not all transactions presently contain the required routing information.)

TRANSLATION TABLE CONSTRUCTION

- Use the translation tables developed for and tested by the LGN prototype test as long as they remain consistent with the transaction definitions. Additional debugging of these tested tables should be minimal.

  Alternative: Of course, new tables that work equally well could be constructed. (Not recommended, because (1) these would require extensive development and testing similar to that which the prototype versions have already passed and (2) the prototype tables benefited from close ties with the analysts involved in creating the EDI/DLSS equivalences.)

PROTOTYPE TRANSLATION SOFTWARE

- Develop translating software that is table-driven; that is, make it subordinate to the control of a table and employ translation logic (rules)

[4]Recall that enhancements to EDI transactions refer to extra-DLSS data contained in the EDI transactions for which no DLSS equivalent exists.

totally embedded within tables. Translating in either direction, the controlling table prescribes the format into which the data are to be transformed. Table-driven software, once developed and debugged, allows changes to transactions through table modifications, without altering the software.

- Require that the table-driven translation software compile the EVAL-tables into executable program steps as part of the software initialization (compilation).

  Alternative 1: Purchase off-the-shelf table-driven software for translation between DLSS and EDI formats. (This is not yet a viable alternative, since no commercial software presently exists to accommodate the deciphering of DLSS-encoded data.)

  Alternative 2: Create DLSS-specific translating software, or translate via artificial intelligence techniques. (Either would work, but not recommended, because (1) specific translation logic would permit few changes in transactions without requiring the translator to be reprogrammed and (2) artificial intelligence techniques need to mature further to be a viable alternative to established table-driven translation methods.)

  Alternative 3: Provide EVAL-tables as another control-type table performing look-up similar to that performed by the primary control table. (Not recommended, because creating EVAL-tables consisting of look-up functions instead of executable routines could slow the translation processing time dramatically. To compensate, a commensurate increase in speed and capacity would be required in the LGN to match the processing rate of the recommended approach.)

- Augment the basic translation-table processing of the prototype via a pre-process module. Examples of required pre-processing include (1) a filter that selects the appropriate translation table on the basis of the DLSS transaction's DIC and (2) the organization of multiple images into a sequence suitable for translation.

  Alternative: Contain the translation process within a single comprehensive translation software package. (Not recommended, because (1) the pre-processing requirement is functionally distinct from the actual format conversion process, (2) adding DLSS-specific criteria complicates rather than simplifies the translation software, and (3) including transaction-related rules in the translation software undercuts the benefits derived from a table-driven (i.e., table-interpreting) approach.)

- Develop and include in the LGN additional non-table functions such as operational control parameters and translation error messages. Non-table

functions include all "housekeeping" activities normal to operational software.

APPENDIX A

THE PROTOTYPE TEST PLAN

THE PROTOTYPE TEST PLAN

A prototype test plan, intended as a guideline for prototype testing procedures, was developed before actual testing began. Overall, the field test was conducted consistently with that plan. This appendix summarizes the planned testing activities and points out deviations from them by comparing actual testing procedures against excerpts from the test plan.

THE FIELD TEST PLAN

The Modernization of Defense Logistics Standard Systems (MODELS) prototype test was performed between FY88 and FY90 and was arranged in three overlapping stages (also called phases). Each stage consists of two steps. The following lists the originally planned stages and steps:

Stage 1: MODELS Testbed Integration

Step 1: Test MODELS translation software and X.25 telecommunications network interface in testbed between the Logistics Management Institute (LMI) and the Defense Automatic Addressing System Office (DAASO).

Step 2: Introduce electronic data interchange (EDI) test transactions into the Defense Automatic Addressing System (DAAS)-III processor.

Stage 2: Test Site Integration and Testing

Step 3: Install test site logistics gateway node (LGN) in testbed; exercise MODELS translation software and X.25 telecommunications interface between test site and DAASO.

Step 4: Parallel-test transaction exchange between test sites.

Stage 3: Live Transaction Testing

Step 5: Test live transaction exchange in the testbed, end-to-end between test sites.

Step 6: Test exchange of transactions with extended capabilities among multiple test sites.

Because the following discussion is based on and contains excerpts from the MODELS Prototype Field Test Plan established prior to the test, the future tense from that plan is preserved. Throughout the test, procedures were adapted to accommodate unplanned exigencies encountered, to react to test findings, and to make practical adjustments. Comments are provided as an explanation when actual procedures deviated from the plan.

Stage 1: MODELS Testbed Integration

In Stage 1, the MODELS testbed will be built and tested with EDI transactions exchanged between LMI and DAASO. After the testbed is built, it will be used to process logistics transactions from selected test sites and to integrate DAAS functions with the MODELS test.

COMMENT: The proposed integration with the Defense Logistics Standard Systems (DLSS) processing system at DAASO was not implemented. Instead, the central LGN (CLGN) at LMI served as surrogate for the DAAS CLGN. A discussion of the intended role of DAASO in the test plan is included because (1) all processing planned for DAASO was performed or emulated at the LMI CLGN and (2) comparable test steps will be required at DAASO in a follow-on pilot test.

Step 1: Test MODELS translation software and X.25 telecommunications network interface in testbed between LMI and DAASO.

The X.25 telecommunications network for passing test data between LMI and DAASO will be installed. At DAASO, the network will interface with a Gould 9050 computer acting as a Defense Data Network (DDN)-to-Ethernet gateway; at LMI, the network will interface via a UNIX network controller. If DDN circuits are not available, alternative telecommunications may be used in the testbed.

Compaq 386 microcomputers will furnish prototype LGN capabilities at LMI and DAASO. The LGNs will provide network connectivity and will contain the translation software. A CLGN established at DAASO will support the following activities:

- Translation from EDI to DLSS (80-column) format to allow interface with the DAAS-III processing module
- Translation from DLSS to EDI format before data are transmitted to the destination site
- Assurance of DAAS-to-destination data transfer.

Step 2: Introduce EDI test transactions into the DAAS-III processor

The DAAS-III module will be coupled with the MODELS testbed to permit DAAS processing of test transactions. Test transactions will (1) originate in 80-column format at LMI, (2) be translated into EDI format, and (3) pass via X.25 protocol over DDN to the CLGN. Transactions arriving at the CLGN will be placed in a queue to await retranslation from EDI to 80-column format.

The CLGN translation software will convert EDI transactions to 80-column format and place them in an outbound queue to DAAS-III. They will enter the DAAS-III processor via the Simple Mail Transfer Protocol through a Gould computer linked to the CLGN by Ethernet and a HYPERchannel network. The Gould computer will pass the test transactions into the DAAS-III editing, routing, and logging modules.

After DAAS processing, the 80-column test transactions will be routed from DAAS-III by HYPERchannel file transfer to the Gould computer. The Gould computer will route them via Ethernet to the CLGN, which will convert them back to EDI format and return them to LMI via the X.25 telecommunications network. The LGN at LMI will receive and translate them from EDI to DLSS format.

This transaction processing path for Step 2 will show initial translation and transmission capabilities, integration of DAAS functions with EDI data, and integration of new MODELS capabilities.

Stage 2: Test Site Integration and Testing

Expanding on the Stage 1 testbed, LMI will assist selected logistics sites in implementing a MODELS translation capability and exchanging EDI transactions via the testbed. This stage will provide translation and transmission of EDI transactions, further utilization of DAAS functions, initial exploration of handling new MODELS capabilities, and a basis for implementation planning.

Step 3: Install test site LGN in testbed; exercise MODELS translation software and X.25 telecommunications interface between test site and DAASO

Compaq 386 microcomputers will furnish prototype LGN capabilities at the test sites. The LGNs will provide network connectivity and will contain the translation software.

Testing for this step will commence with two participating logistics sites: Wright-Patterson Air Force Base, Dayton, Ohio (a retail activity), and Defense Construction Supply Center, Columbus, Ohio [an inventory control point (ICP)]. Those locations will become groundbreaking sites for testing EDI data transfer. The requisition transaction will be tested first.

This planned step is analogous to the LMI and DAASO interface test (Step 1); the data will not be introduced into the DAAS-III systems. LMI will provide each site with a prototype LGN containing telecommunications and MODELS translation software and will ensure that communications software is tailored to site requirements.

The site will provide any programming to expand 80-column data into extra-DLSS data for transactions to be sent. For example, a site may modify its system software for a DLSS-format requisition to add weapon system data in a trailer card. Those data will be downloaded and passed to the MODELS translator. The entire transaction, both DLSS and extra-DLSS components, will be translated into EDI format and packaged for X.25 transmission over the telecommunications network. Transactions in X.25 format will be conveyed from the LGN over an Ethernet local area network to the long-haul telecommunications network.

COMMENT: Extra-DLSS (enhanced EDI) data were not processed during the field test. The means by which enhanced data components may be handled in a pilot test or in an operational system are discussed under Translation Table Construction in Chapter 4 of this volume.

Step 4: Parallel-test transaction exchange between test sites

This step is analogous to Step 2, except that it will include a logistics activity test site. Step 4 will be a parallel test of DLSS transactions and EDI-formatted counterparts. Transactions will include requisitions, follow-ups, requisition modifiers, and cancellations. Copies of live data will be passed through the network in EDI format in parallel with live data in DLSS format on Automatic Digital Network (AUTODIN); actual logistics operations will use the DLSS format.

COMMENT: No live data were exchanged via the testbed during the prototype test. This part of the field test must await (1) standardization of LGN and host interconnection, (2) EDI processing capabilities at DAASO, and (3) test sites that are amenable to live data testing. The remaining Stage 3 plan for live testing is included for completeness and as a guide for its accomplishment in a follow-on pilot test.

Stage 3: Live Transaction Testing

In this last stage of the test, more test sites will be added, and some will begin to exchange live EDI transactions via the testbed. Stage 3 will (1) complete testing of translation and transmission of EDI transactions and integration with DAAS functions, (2) further explore the integration of new MODELS capabilities, and (3) provide a basis for implementation planning.

Step 5: Test live EDI transaction exchange in the testbed, end-to-end between test sites

In Step 5, live testing of interfaces supporting EDI-to-DLSS, DLSS-to-EDI, and EDI-to-EDI transactions will begin. The first test will include a requisitioner and an ICP. Under that test, selected live data will be sent between test sites in EDI format.

DAAS support of end-to-end transmission with logistics sites using different standards will be tested in this step. Before the step begins, DAASO must have developed the ability to selectively route outbound transactions through its AUTODIN front-end (CDC 1700) computer or through the CLGN to DDN, depending on whether the destination site (1) is or is not part of the prototype test and (2) uses DLSS format or EDI format. This step will demonstrate that DLSS and EDI transactions can coexist, as can EDI-capable and DLSS-only sites, and that both pairs can operate within a MODELS environment. At the time of the Stage 3 test, some or all of the DAAS-III systems may have been replaced by Gould 9050/Zenith 248 computers or follow-on systems.

Step 6: Test exchange of transactions with extended capabilities among multiple test sites

Finally, after live transaction testing has been completed successfully at a site, extra-DLSS data and additional EDI transactions (e.g., automated Supply Discrepancy Report transactions), as defined by the Functional Working Group, may be added to the test. If the site has modified its internal entry and processing systems to send and receive EDI transactions directly, then the site host computer can be connected to the network and the translation software bypassed, allowing direct EDI transaction entry, receipt, and processing.

Additional test sites may be identified and added to the test, on an ad hoc basis, to demonstrate a particular scenario or activity that was not covered by any of the original test sites. In particular, more complete testing of the transportation transactions will occur during this step.

COMMENT: Before transportation data could be included in the prototype test, the test was terminated.

APPENDIX B

FIELD TEST EXAMPLES

FIELD TEST EXAMPLES

NAVAL SUPPLY CENTER, NORFOLK, VIRGINIA

The supply center's host is a Tandem TXP; the logistics gateway node (LGN) connected to it via a 9600-baud dial-up line. During LGN boot-up, a terminal emulator program, known as PCT (PC-Terminal), is loaded not as a device driver but as a separate process under DESQview. The PCT program lets the LGN emulate a Tandem terminal and conduct a host session, but it does not have any file transfer capability.

To download a file from the host, someone at the site must take a one-time manual step of exiting from the PCT program into the disc operating system and invoking the Information Xchange Facility (IXF) file transfer program.

The batch program also allows the user to call IXF manually; this capability was useful, since the schedule for downloads was variable. IXF must be called from within the same DESQview process as PCT, which runs in the background. Consequently, the IXF software cannot be invoked when needed as part of the download script process.

To be consistent with other sites, the main local interface module still calls a dummy download script process, which immediately returns without doing any real work. The real work (the download) is controlled by the constantly running PCT/IXF process.

TROOP SUPPORT COMMAND, ST. LOUIS, MISSOURI

The host at Troop Support Command (TROSCOM) is a Sperry 5080 minicomputer running under UNIX. It is accessed via a shared modem pool known as PACX. The wide area network (WAN) interface module also connects to t'.e WAN through PACX. The LGN has a single connection to PACX, so contention between the two modules must be managed.

Contention is controlled by using a file as a semaphore. Before PACX is accessed, a check is made for the semaphore file. If it exists, then PACX is in use by the LGN. If repeated checks continually reveal that the semaphore file exists, the module eventually gives up. If there is no semaphore file, the module creates one before accessing PACX and deletes it after disconnecting from PACX. Both the local interface and WAN interface modules at TROSCOM have this logic built in.

The TROSCOM local download script module invokes Crosstalk Mk.4 to control the host session. The session starts with a PACX dialog resulting in a connection to the Sperry minicomputer. The next step is to log on the Sperry minicomputer and begin a dialog with Kermit, which is resident on the host. The LGN then downloads the host file (if it exists) and shuts off Kermit. After the download attempt, the LGN logs off the Sperry minicomputer and disconnects from PACX.

The communications environment at TROSCOM was sufficiently atypical to warrant special code in the WAN interface module for handling these unique conditions. Further, the differences at TROSCOM could not be accommodated feasibly using table-driven methods. Therefore, a program compilation flag was used so that code specific to TROSCOM was compiled if the TROSCOM flag existed. For the LGN prototype, all communications at TROSCOM (local and wide-area) traversed the PACX shared modem pool. The following summarizes the special processing required:

- The local interface subsystem and the WAN interface subsystem share the same port for connecting to PACX. A file is employed as a semaphore to indicate which subsystem has use of the port.
- Initializing a serial communications session with PACX differs from initializing a session with a modem; the wake-up sequence is different.
- Instead of using standard Hayes modem commands, PACX-specific commands are used.
- All checks for the presence of the "Carrier Detect" signal are bypassed.
- The hangup sequence is different from that for a modem. Special PACX codes are sent, and then the "Data Terminal Ready" signal is dropped.
- When receiving incoming data during call initialization, the parity bit, which PACX applies unconditionally, is ignored. Normally, the WAN interface module does not use a parity bit.

APPENDIX C

COST MODEL RESULTS

COST MODEL RESULTS

```
                              ANNUALIZED LGN COSTS

                SCENARIO:Current Centralized Operation (Commercial Packet)
                  YEARS:10

   SCENARIO INPUTS              NUMBER        UNIT COST      TOTAL COST
                              ------------------------------------------

EQUIPMENT CHARGES:
NUMBER OF TYPE 1 SITES              15       $23,433.59     $351,503.85
          TYPE 2 SITES               0       $24,769.00           $0.00
          TYPE 3 SITES               0       $33,084.20           $0.00
                                                        ---------------
TOTAL EQUIPMENT CHARGES                                     $351,503.85


                                NUMBER    TOTAL COST/YR      TOTAL COST
                              ------------------------------------------
TOTAL MAINTENANCE CHARGES: From Above        $23,610.60     $236,106.00

TOTAL LINE CHARGES:
                                                        ---------------
NUMBER DED 9.6 KBPS CKTS            15       $14,400.00   $2,160,000.00

PACKET GROWTH PER YEAR              5%

YEAR 1 KILOPACKETS-INITIAL       51111      $143,110.80
YEAR 2 KILOPACKETS               53667      $150,266.34
YEAR 3 KILOPACKETS               56350      $157,779.66
YEAR 4 KILOPACKETS               59167      $165,668.64
YEAR 5 KILOPACKETS               62126      $173,952.07
YEAR 6 KILOPACKETS               65232      $182,649.68
YEAR 7 KILOPACKETS               68494      $191,782.16
YEAR 8 KILOPACKETS               71918      $201,371.27
YEAR 9 KILOPACKETS               75514      $211,439.83
YEAR 10 KILOPACKETS              79290      $222,011.82
                                                        ---------------
TOTAL PACKET CHARGES                                      $1,800,032.26

DIAL-IN CHARGES:
PERCENTAGE DIAL-IN PACKETS       0.00%
TOTAL KILOPACKETS                    0
TOTAL MINUTES (1.2 Kbps)             0
                                                        ---------------
TOTAL COST                                                        $0.00


                                                        +--------------
TOTAL SCENARIO LIFE CYCLE COST                          | $4,547,642.11
                                                        +--------------
```

```
                         ANNUALIZED LGN COSTS

          SCENARIO:Current Centralized Operation (DDN)
            YEARS:10

      SCENARIO INPUTS          NUMBER        UNIT COST       TOTAL COST
                               ------------------------------------------

EQUIPMENT CHARGES:
NUMBER OF TYPE 1 SITES             15       $23,433.59      $351,503.85
          TYPE 2 SITES              0       $24,769.00            $0.00
          TYPE 3 SITES              0       $33,084.20            $0.00
                                                         ---------------
TOTAL EQUIPMENT CHARGES                                     $351,503.85


                               NUMBER    TOTAL COST/YR       TOTAL COST
                               ------------------------------------------
TOTAL MAINTENANCE CHARGES: From Above       $23,610.60      $236,106.00

TOTAL LINE CHARGES:
                                                         ---------------
NUMBER DED 9.6 KBPS CKTS           15       $12,600.00    $1,890,000.00

PACKET GROWTH PER YEAR              5%

YEAR 1 KILOPACKETS-INITIAL      51111       $68,999.85
YEAR 2 KILOPACKETS              53667       $72,449.84
YEAR 3 KILOPACKETS              56350       $76,072.33
YEAR 4 KILOPACKETS              59167       $79,875.95
YEAR 5 KILOPACKETS              62126       $83,869.75
YEAR 6 KILOPACKETS              65232       $88,063.24
YEAR 7 KILOPACKETS              68494       $92,466.40
YEAR 8 KILOPACKETS              71918       $97,089.72
YEAR 9 KILOPACKETS              75514      $101,944.20
YEAR 10 KILOPACKETS             79290      $107,041.41
                                                         ---------------
TOTAL PACKET CHARGES                                        $867,872.70

DIAL-IN CHARGES:
PERCENTAGE DIAL-IN PACKETS      0.00%
TOTAL KILOPACKETS                   0
TOTAL MINUTES (1.2 Kbps)            0
                                                         ---------------
TOTAL COST                                                        $0.00


                                                         +--------------
TOTAL SCENARIO LIFE CYCLE COST                           | $3,345,482.55
                                                         +--------------
                         ANNUALIZED LGN COSTS

           SCENARIO:Current Centralized Operation (Dedicated Line)
             YEARS:10

     SCENARIO INPUTS          NUMBER        UNIT COST      TOTAL COST
                             -----------------------------------------

EQUIPMENT CHARGES:
NUMBER OF TYPE 1 SITES            15       $23,433.59     $351,503.85
          TYPE 2 SITES             0       $24,769.00           $0.00
          TYPE 3 SITES             0       $33,084.20           $0.00
                                                         ---------------
TOTAL EQUIPMENT CHARGES                                   $351,503.85


                              NUMBER    TOTAL COST/YR      TOTAL COST
                             -----------------------------------------
TOTAL MAINTENANCE CHARGES: From Above      $23,610.60     $236,106.00

TOTAL LINE CHARGES:
                                                         ---------------
NUMBER DED 9.6 KBPS CKTS          15                    $1,866,916.80

PACKET GROWTH PER YEAR            0%

YEAR 1 KILOPACKETS-INITIAL         0            $0.00
YEAR 2 KILOPACKETS                 0            $0.00
YEAR 3 KILOPACKETS                 0            $0.00
YEAR 4 KILOPACKETS                 0            $0.00
YEAR 5 KILOPACKETS                 0            $0.00
YEAR 6 KILOPACKETS                 0            $0.00
YEAR 7 KILOPACKETS                 0            $0.00
YEAR 8 KILOPACKETS                 0            $0.00
YEAR 9 KILOPACKETS                 0            $0.00
YEAR 10 KILOPACKETS                0            $0.00
                                                         ---------------
TOTAL PACKET CHARGES                                             $0.00

DIAL-IN CHARGES:
PERCENTAGE DIAL-IN PACKETS     0.00%
TOTAL KILOPACKETS                  0
TOTAL MINUTES (1.2 Kbps)           0
                                                         ---------------
TOTAL COST                                                       $0.00


                                                         +--------------
TOTAL SCENARIO LIFE CYCLE COST                           | $2,454,526.65
                                                         +--------------
```

APPENDIX D

ACRONYMS

ACRONYMS

ADP	=	automated data processing
ANSI	=	American National Standards Institute
ASC	=	Accredited Standards Committee
AUTODIN	=	Automatic Digital Network
CLGN	=	central logistics gateway node
DAAS	=	Defense Automatic Addressing System
DAASO	=	Defense Automatic Addressing System Office
DDN	=	Defense Data Network
DIC	=	document identification code
DLA	=	Defense Logistics Agency
DLSS	=	Defense Logistics Standard Systems
DLSSD	=	Defense Logistics Standard Systems Division
DoD	=	Department of Defense
DOS	=	disk operating system
EDI	=	electronic data interchange
Kbps	=	kilobits per second
LAN	=	local area network
LGN	=	logistics gateway node
Mhz	=	mega-hertz
MILSBILLS	=	Military Standard Billing System
MILSCAP	=	Military Standard Contract Administration Procedures
MILSPETS	=	Military Standard Petroleum System Procedures
MILSTAMP	=	Military Standard Transportation and Movement Procedures

MILSTRAP	=	Military Standard Transaction Reporting and Accounting Procedures
MILSTRIP	=	Military Standard Requisition and Issue Procedures
MODELS	=	Modernization of Defense Logistics Standard Systems
MS-DOS	=	MicroSoft Disc Operating System
OSD	=	Office of the Secretary of Defense
PKE	=	public key encryption
SDR	=	Supply Discrepancy Report
TROSCOM	=	U.S. Army Troop Support Command
TWG	=	Technical Working Group
WAN	=	wide area network